韓國 컴퓨터情報學會 論文誌
第11卷 第3號, 2006. 7.
2006-11-3-1-14

# 이동 에이전트의 효율적 이주를 위한 최적 경로 탐색

김 광 종*, 고 현**, 김 영 자***, 이 연 식****

# Optimal Path Searching for Efficient Migration of Mobile Agent

Kwang-jong Kim*, Hyun Ko **, Young-ja Kim***, Yon-sik Lee****

## 요 약

본 논문에서는 자율적인 작업 처리 능력을 가진 이동 에이전트의 효율적인 이주를 위해 네트워크 트래픽 감지를 이용한 최적의 이주 노드 경로 탐색 및 최적 노드로의 경로조정 기법과 노드 재지정 기법을 제시한다. 기존의 이동 에이전트는 사용자로부터 다양한 작업을 부여받거나 혹은 에이전트의 작업 처리 결과량이 매우 많을 경우, 에이전트의 크기 증가로 인해 네트워크 과부하 및 트래픽을 가중시킬 수 있다. 또한, 많은 네트워크 트래픽이 발생한 시점에서 사용자가 지정한 라우팅 스케줄(Routing Schedule)에 따라 이주를 수행하면 많은 노드 순회 시간 비용이 소요됨은 물론 통신망 결손 및 노드 장애와 같은 특정한 상황에 능동적으로 대처할 수 없다. 따라서 본 논문에서는 많은 네트워크 트래픽 발생 경로나 시점에서의 에이전트 이주 시 사용자에 의한 수동적 라우팅 스케줄 지정 방식에서 탈피하여 네트워크 트래픽 감지를 이용한 목적노드까지의 최적 경로 탐색을 통해 자동적인 경로 조정을 수행하는 에이전트의 이주 기법을 제시한다. 최적 이주 경로 탐색 및 조정 기법은 네트워크 트래픽 경로를 회피하는 방식으로 이동 에이전트의 순회 작업 처리 시간을 단축시키고 분산 노드에 대한 이주 신뢰성을 보장해준다.

## Abstract

In this paper, we propose the optimal migration path searching method including path adjustment and reassignment techniques for an efficient migration of mobile agent which has an autonomous ability for some task processing. In traditional agent system, if the various and large quantity of task processing were required from the users, it could be derive excessive network traffic and overload due to increasing the size of agent. And also, if an agent migrates from node to node according to routing schedules specified by the user at which the most of network traffic occurs, it could not actively cope with particular situations such as communication loss and node obstacles and required much cost for node

• 제1저자 : 김광종
• 접수일 : 2006.06.13, 심사일 : 2006.06.30, 심사완료일 : 2006.07.16
* 군산대학교(이학박사), ** 군산대학교 대학원(박사과정), *** 인천기능대학 교수, **** 군산대학교 교수
※ 이 논문은 한국과학재단 특정기초연구 (R01-2004-000-10946-0)지원으로 수행되었음

traversal. Therefore, this paper presents the migration method of agent which can try to adjust and reassign path tothe destination automatically through the optimal path search using by network traffic perception in stead of the passive routing schedule by the user. An optimal migration path searching and adjustment methods ensure the migration reliability on distributed nodes and reduce the traversal task processing time of mobile agent by avoiding network traffic paths and node obstacles.

▸ Keyword : 이동 에이전트(Mobile Agent), 서버 푸시 에이전트(SPA), 클라이언트 푸시 에이전트(CPA)

## I. 서 론

가최근 새로운 방식의 분산 컴퓨팅 메커니즘들 중 많은 연구가 지속적으로 이루어지고 있는 이동 에이전트 기술은 통신망의 점유와 과부하를 감소시켜 분산된 시스템들이 가진 다양한 문제들을 효과적으로 해결할 수 있게 하는 분산 시스템의 새로운 패러다임이다[1,2,3,5]. 현재 이동 에이전트 기술은 정보 검색 및 분산처리, 이동 컴퓨팅, 전자상거래, 네트워크 관리 등 다양한 분야에 응용이 확산되고 있다. 그러나 다양한 분야에서 사용되는 있는 기존의 이동 에이전트 시스템들은 사용자로부터 복잡하고 다양한 작업을 부여받거나 혹은 노드들을 이주중인 에이전트의 작업 처리 결과가 매우 많을 경우, 에이전트의 크기를 증가시켜 네트워크 트래픽을 가중시키게 되고, 이로 인해 노드 이주를 통한 순회 작업 처리 시 보다 많은 시간이 소요된다. 또한, 대부분의 에이전트 시스템들은 수동적 라우팅 스케줄에 따라 노드 이주를 수행함으로써 통신망 결손이나 노드 장애와 같은 특정한 상황에 동적으로 대처할 수 없으며 신뢰성 있는 이주 보장을 제공하지 않는다[4]. 따라서 에이전트 크기에 따른 네트워크 트래픽 가중 문제를 해결하여 에이전트의 순회 작업 처리 시간을 최소화시키고 노드 간 에이전트의 이주 신뢰성을 보장하기 위한 새로운 에이전트 이주 기법이 필요하다. 이에 본 논문에서는 많은 네트워크 트래픽 발생 시점에서의 에이전트 이주 시 사용자에 의한 수동적 라우팅 스케줄 지정 방식에서 탈피하여 네트워크 트래픽 감지를 이용한 목적 노드까지의 최적 경로 탐색을 통해 자동적인 경로 조정을 수행하는 에이전트의 이주 기법을 제시한다. 또한, 최적 경로 탐색 및 경로 조정을 위해서 필요한 네트워크 트래픽 측정 방법에 대해 기술하고, 트래픽 감지 과정에서의 통신망 결손이나 노드 장애에 따른 신뢰성 있는 이주 보장을 위해 노드 재지정 기법을 제시한다.

본 논문의 구성은 2장에서 관련연구로서 기존 이동 에이전트의 이주 방법과 최적 경로 탐색을 지원하는 이동 에이전트 모델에 대해 설명하고, 3장에서는 효율적인 에이전트의 이주를 지원하기 위해 네트워크 트래픽을 측정하는 방법에 대해 기술한다. 4장에서는 최적의 이주 경로 탐색 및 경로 조정 기법과 물리적 장애에 따라 이주 노드를 재조정하는 기법을 제시한다. 마지막 5장에서는 결론 및 향후 연구 방향에 대해 기술한다.

## II. 관련연구

기존에 개발된 대부분의 이동 에이전트 시스템들은 이동 에이전트의 이주 전략과 관련하여 특별한 이주 보장 정책을 지원하지 않거나 이를 지원한다 해도 대부분 노드 혹은 호스트 결점에 대한 이주 신뢰성을 보장하기 위한 이주 정책들만을 제공하고 있다[5,6,7,8,9,10,12]. 노드 결손이나 호스트 장애 또는 데이터베이스 등 정보 서비스의 부재 시 이동 에이전트는 무한 대기 상태나 고아(orphan) 상태에 빠질 수 있고 혹은 파괴되어 쓰레기로 처리될 수 있다[4]. 이러한 이동 에이전트의 이주 시 발생 가능한 노드 결손 및 호스트 장애에 대처하기 위해, JAMAS[12]에서는 경로 재조정과 후위 복구 기법을 제안하여 이동 에이전트의 이주를 보장할 수 있도록 하였으며, 이동 에이전트의 이주 정책[4,11]에서도 이주하는 노드의 결점에 대처하기 위한 프로토콜을 제시하였다. 또한, Voyager[10]는 이동 에이전트의 생명주기를 위한 5가지 정책을 지원하고, Mole[4]는 이동 에이전트에 대한 고아 찾기와 성공적인 종료를 위한 그림자(shadow) 프로토콜을 제공한다. 한편, 이동 에이전트의 이주 노드에 대한 라우팅 스케줄 지정과 관련하여 대부분의 이동 에이전트 시스템은 Aglets[6,7]에서와 같이 이동 에이전트가 고정된 순서대로 노드를 방문하고 방문한 노드에

서 검색한 데이터를 계속 누적시키면서 이동하는 단순 구조를 가지고 있다. 이는 복잡하고 다양한 사용자 요구를 부여받은 에이전트의 작업 수행에 있어 노드 재지정을 필요로 하는 경우나 통신망 결손 및 노드 장애와 같은 특정한 상황에 동적으로 대처할 수 없는 문제가 있다. 이와 같이 현재에는 이동 에이전트의 이주 전략과 관련하여 노드 혹은 호스트의 결점이 발생하였을 때 이주 신뢰성을 보장하기 위한 연구가 주류를 이루고 있으며, 최소의 네트워크 소요시간을 갖도록 이동 에이전트의 이주 계획을 세우는 알고리즘에 대한 연구는 아직 제시되지 않았다.

## III. 노드 이주와 네트워크 트래픽

### 3.1 지능형 경로 탐색 이동 에이전트 모델

이동 에이전트 모델은 네트워크 트래픽 감지를 통해 현재의 노드에서 라우팅 테이블의 각 목적 노드들 중 이주 경로가 최적인 노드를 판단하여 이주를 수행함으로써 네트워크 상의 모든 목적 노드에 대한 순회 작업 처리 시 소요되는 시간 비용을 최소화한 에이전트 프레임웍이다. (그림 1)은 능동적인 최적 경로 탐색을 통해 에이전트의 작업을 보다 효율적으로 처리하는 경로 탐색 이동 에이전트 모델이다.

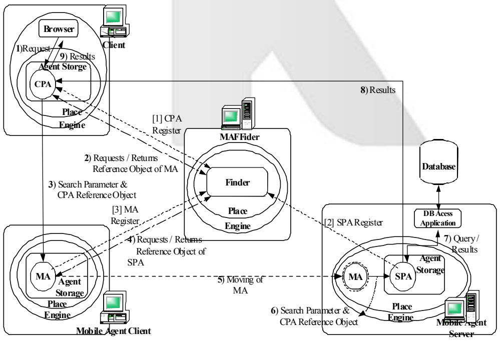


그림 1. 경로 탐색 이동 에이전트 모델
Fig 1. Path Searching Mobile Agent Model

이동 에이전트 모델은 이동 에이전트 수행 패턴 중 하나인 Locker 패턴[13]을 응용한 것으로, 이동 에이전트가 방문하는 각각의 노드에서 획득한 결과를 임의의 영역에 보관하는 대신 추가로 개발하는 푸시 에이전트를 이용하여 이를 독자적으로 클라이언트에 전송하는 형태이다. 또한, 기존의 사용자에 의한 수동적 라우팅 스케줄 지정 방식에서 탈피하여 (그림 1)에서의 MAFFinger를 이용한 능동적 라우팅 스케줄 지정 방식을 사용함으로써 이동 에이전트가 과다한 트래픽 발생 경로를 회피하여 최적 경로를 통해 이주를 수행할 수 있도록 지원한다.

한편, 지능형 경로 탐색 이동 에이전트 모델은 <표 1>에서와 같이 이동 에이전트 플랫폼과MAFFinder, 이동 에이전트, 푸시 에이전트로 구성된다. <표 1>은 이동 에이전트 모델의 각 구성 요소와 그 기능들에 대한 설명이다.

표 1. 경로 탐색 이동 에이전트 모델의 구성 요소
Table 1. Components of Path Searching Mobile Agent Model

| 구성 요소 | |
|---|---|
| 이동 에이전트 플랫폼 | 에이전트가 주어진 작업을<br>전트와 각 에이전트들의 상<br>템으로의 에이전트 이주 및 |
| MAFFinder | 분산된 노드의 위치 투명성<br>위해 네이밍 서비스를 이용<br>정확한 정보 서비스를 위한 |
| 이동 에이전트 | CORBA의 분산 객체 간 통신<br>클라이언트 객체를 이동성<br>워크 부하 및 순회 작업 처리 |
| 푸시 에이전트 | 서버 푸시 에이전트는 기존<br>해온 에이전트의 호출을 받<br>트 푸시 에이전트는 서버 푸<br>과정을 통해 중복 데이터를 |

### 3.2 이동 에이전트의 노드 이주

에이전트의 순회 작업 처리에 있어 노드 간 이주 시간은 네트워크의 상태에 따라 에이전트의 구조와 에이전트의 노드 이주 방식이 큰 영향을 미친다. 먼저, 에이전트 구조와 이주 시간과의 관계에 있어, 에이전트 크기는 네트워크의 부하 발생 요인으로 작용하여 이동 에이전트의 이주 시간을 많이 걸리게 하거나 적게 걸리게 한다. 기존의 이동 에이전트는 에이전트 객체 내부에 작업 실행 모듈을 포함하고 있으며, 에이전트가 처리한 작업 결과를 직접 가지고 이주를 수행한다. 이는 에이전트의 크기를 증가시켜 네트워크 부하를 발생시키게 되고, 이로 인해 많은 에이전트의 노드 이주 시간을 소요하게 한다.

한편, 이동 에이전트의 크기가 일정할 경우, 이주 경로 내에 발생한 네트워크 트래픽의 양에 따라 에이전트의 이주 시간이 많이 걸리거나 적게 걸릴 수 있다. 기존 에이전트 시스템의 이주 정책들은 대부분 통신망 결손이나 노드 장애와 같은 물리적인 요인에 대비한 이주 신뢰성만을 보장하기 위한 방법들을 제공하고 있을 뿐 에이전트의 이주 시간을 단축시켜 보다 효율적으로 작업을 처리하기 위한 방법들은 지원하지 않는다. 따라서 네트워크 트래픽에 따른 이동 에이전트의 노드 순회 시간을 단축하기 위한 방법이 필요하다.

이와 같이 기존 이동 에이전트 시스템들이 가진 노드 이주 시의 문제점들을 해결하기 위해서는 에이전트의 크기를 최소화하고 네트워크 트래픽에 적절하게 대응하여 이주를 수행해야 한다. 먼저, 에이전트의 크기를 최소화하기 위해

CORBA의 분산 객체 형태로 이동 에이전트를 호출 모듈 객체와 실행 모듈 객체로 분리하고, Locker 패턴의 적용을 통해 에이전트의 실행 모듈을 포함하는 푸시 에이전트를 생성하여 이동 에이전트의 작업 처리 결과를 사용자에게 독자적으로 전달하도록 한다. 이러한 이동 에이전트의 수행 패턴은 (그림 2)와 같다.

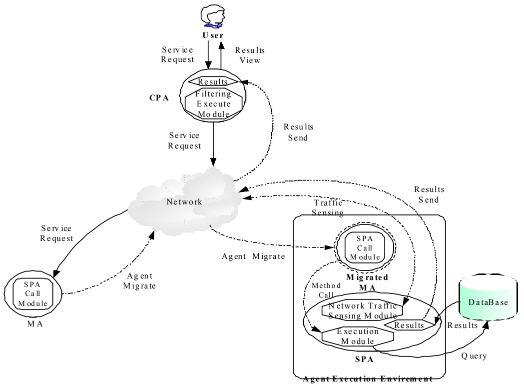


그림 2. 이동 에이전트의 수행 패턴
Fig 2. Execution Pattern of Mobile Agent

또한, 네트워크 트래픽으로 인한 이주 지연 시간 문제는 4장에서 제시하는 최적 이주 경로 탐색 및 조정 기법과 노드 재지정 기법을 이용하여 이동 에이전트가 최적의 이주 경로로 이주를 수행할 수 있도록 함으로써 이주 시간을 최소화한다.

## 3.3 네트워크 트래픽 측정

네트워크 트래픽 측정은 한 노드에서 목적 노드들까지의 경로들 중 최적 경로를 찾기 위해 네트워크 부하를 유발하여 에이전트의 이주를 지연시키는 트래픽 크기를 측정하는 것이다. 네트워크 트래픽 측정 방법은 현재의 노드에서 목적 노드까지의 일정 단위 경로에서 하나의 패킷을 목적 노드에 보내고 다시 리턴되어 돌아오는 시간을 체크함으로써 시간에 따른 트래픽 크기를 측정할 수 있다. 즉, 일정한 대역폭을 갖는 단위 경로 상에서 트래픽이 발생하지 않는 상태와 불규칙적인 트래픽이 발생하고 있는 상태의 패킷 리턴 시간을 측정하고, 두 상태에서 측정된 시간값의 차이를 계산하여 트래픽 크기로 표현한다. 패킷의 리턴 시간, 즉 패킷 왕복 시간을 측정하기 위해서는 Ping 프로그램을 사용할 수 있다. Ping 프로그램은 패킷 왕복 시간은 물론 패킷의 분실 여부 및 N 개의 패킷에 대한 평균 왕복 시간과 최대 또는 최소의 왕복 시간을 측정할 수 있다. (그림 3)은 Windows의 Ping 프로그램을 Java 소스 상에서 실행하는 알고리즘과 그 실행 결과이다.

```
  Public class Ping{  Public Static void Main(String[]
args){
  Try{ //creation of Ping Process
  Process p = Runtime.getRuntime().exec("ping n 10
     202.31.147.142");
  byte[] msg = new byte[500];
  int len;
  //write msg size through Ping information
InputStream
while((len=p.getInputStream().read(msg))> 0)
  { System.out.print(new String(msg, 0, len));}
  ...
  }catch(Exception e){e.printStackTrace();} } }
```

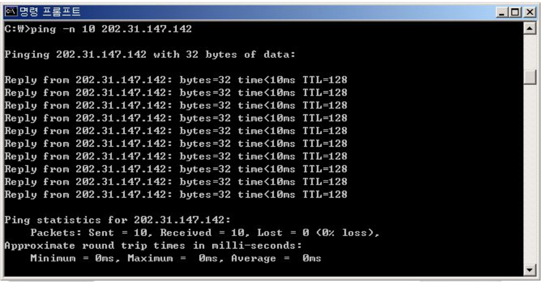


그림 3. Ping 프로그램의 실행 결과
Fig 3. Result of Ping Program

네트워크 트래픽을 측정하기 위한 실험 모델에서는 Ping을 실행하여 얻어진 정보들 중 패킷의 평균 왕복 시간인 Average 값을 추출하여 이용한다. 또한, Ping의 평균 패킷 왕복 시간 측정 시 시간적인 영향을 줄 수 있는 네트워크의 대역폭과 통신 선로의 길이에 대해 동일한 균등 네트워크 환경(Uniform Network)임을 가정하였다. 트래픽 측정을 위한 실험 모델에서는 먼저, 네트워크 트래픽이 0인 상태에서의 패킷 왕복 시간과 패킷 크기와의 관계에 대한 분석이 요구된다. 따라서 다음의 실험을 통해 패킷 왕복 시간과 패킷 크기와의 관계를 분석하고, 이를 이용하여 네트워크의 트래픽을 측정한다.

### 실험 1. 패킷 왕복 시간과 패킷 크기와의 관계 분석

가정 : 네트워크의 대역폭은 100 Mbps 이고, 네트워크 통신 선로의 길이는40m 이다.

조건 : 패킷의 크기를 500 Kbyte 단위로 증가시켜 원격의 두 노드 간 평균 패킷 왕복 시간을 측정한다.

결과 : 평균 패킷 왕복 시간은 패킷의 크기가 0.5 ~ 1 Mbyte인 경우를 제외하고는 각각 500 Kbytes씩 증가할 때마다 1.5m/s ~ 2.0m/s가 지속적으로 증가한다.

<표 2>는 패킷의 크기 증가에 따른 패킷의 평균 왕복 시간값이고, (그림 4)는 평균 패킷 왕복 시간의 변화 추이를 나타낸 그래프이다.

표 2. 패킷 크기 증가에 따른 평균 패킷 왕복 시간
Table 2. The average round trip times of packet according to packet size

| Packet Size(Mbytes) | 0.5 | 1.0 | 1.5 | 2.0 | 2.5 | 3.0 | 3.5 | 4.0 | 4.5 | 5.0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.4 | 3.2 | 3.8 | 4.7 | 5.2 | 6.2 | 7.1 | 8.8 | 9.8 | 10.5 |
| 2 | 1.5 | 2.8 | 3.6 | 4.3 | 5.3 | 6.4 | 7.2 | 8.6 | 10.1 | 10.4 |
| 3 | 1.2 | 2.8 | 3.9 | 4.5 | 5.6 | 6.6 | 6.9 | 9.0 | 9.7 | 9.9 |
| 4 | 1.4 | 3.1 | 3.8 | 4.3 | 5.5 | 6.5 | 6.8 | 8.4 | 9.8 | 10.6 |
| 5 | 1.5 | 3.0 | 4.1 | 4.7 | 5.3 | 6.1 | 7.1 | 8.5 | 9.7 | 10.5 |
| Average Round Trip Time(m/s) | 7 | 14.9 | 19.2 | 22.5 | 26.9 | 31.8 | 35.1 | 43.3 | 49.1 | 51.9 |

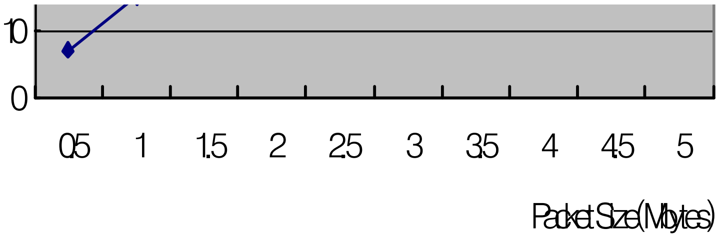


그림 4. 패킷 크기에 따른 평균 패킷 왕복 시간의 변화 추이
Fig 4. Round trip times transition of packet according to packet size

표 3. 패킷 크기에 따른 트래픽 크기값
Table 3. Traffic sizes values according to packet size

| Packet Size(Mbytes) | 0.5 | 1.0 | 1.5 | 2.0 | 2.5 | 3.0 | 3.5 | 4.0 | 4.5 | 5.0 |
|---|---|---|---|---|---|---|---|---|---|---|
| [Traffic] Average Round Trip Time(m/s) | 8 | 16.1 | 21.5 | 24.8 | 29.8 | 34.5 | 38.8 | 46.9 | 52.9 | 56.7 |
| [Nontraffic] Average Round Trip Time(m/s) | 7 | 14.9 | 19.2 | 22.5 | 26.9 | 31.8 | 35.1 | 43.3 | 49.1 | 51.9 |
| Traffic Size(m/s) | 1 | 1.2 | 2.3 | 2.3 | 2.9 | 2.7 | 3.7 | 3.6 | 3.8 | 4.8 |

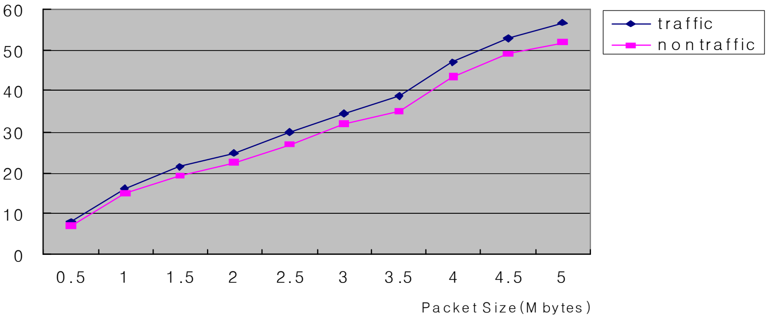


그림 5. 패킷 크기의 증가에 따른 트래픽 크기 그래프
Fig 5. Transition graph for traffic sizes in relation to packet size

### 실험 2. 패킷의 크기에 따른 네트워크 트래픽 측정

가정 : 네트워크의 대역폭은 100 Mbps 이고, 노드 간 거리는 40m 이다.

조건 : 불규칙한 네트워크 트래픽이 발생하고 있는 상황에서, 패킷의 크기를 500 Kbytes 단위로 증가시킬 때 두 노드 간의 경로 상에 발생한 네트워크 트래픽을 측정한다.

결과 : 평균 패킷 왕복 시간은 네트워크 트래픽이 발생하지 않은 상태에서 측정한 시간보다 최소 1.0m/s ~ 5.0m/s가 더 소요된다.

<표 3>은 네트워크 트래픽의 유, 무 상태 시에 측정한 평균 패킷 왕복 시간값의 차를 구해 얻은 트래픽 크기값이고, (그림 5)는 두 상태에서 측정한 평균 패킷 왕복 시간의 변화 추이를 비교하여 패킷의 크기 증가 시 트래픽 크기를 알 수 있도록 나타낸 그래프이다.

## IV. 이동 에이전트의 노드 이주 기법

### 4.1 최적 이주 경로 탐색 및 조정 기법

최적 이주 경로 탐색 및 조정 방법은 이동 에이전트의 이주 시간을 단축시키기 위한 방법으로, 에이전트가 이주할 노드들 중 최소의 네트워크 소요 시간을 갖는 경로를 선택하는 것이다. 즉, 네트워크의 부하 발생 요인인 트래픽을 측정하여 과다한 트래픽이 발생하는 경로는 회피하고 최소의 네트워크 지연을 갖는 경로를 선택하여 보다 효율적인 작업처리가 이루어지도록 하기 위한 방식이다. 이러한 최적 이주 경로 탐색 및 조정 방식의 수행 과정은 다음과 같다.

(1) 이동 에이전트는 사용자로부터 입력받은 검색 키워드를 이용하여 MAFFinder에 등록된 모든 SPA(Server Push Agent) 객체들 중 해당 정보 서비스를 지원할 수 있는 SPA 객체 참조자 리스트(이하 라우팅 테이블)를 전달받는다. 이 때, SPA 객체 참조자 리스트는 검색 적중률이 높은 순으로 되어 있다.

(2) 이동 에이전트는 검색 적중률이 우선인 라우팅 테이블 내의 SPA 객체 참조자를 이용하여 모든 원격 노드에 Ping을 실행한다.

(3) Ping을 실행하여 얻은 정보들 중 각 노드 간 패킷의 평균 왕복 시간인 Average 값과 패킷의 분실 수인 Lost 값을 추출한다.

(4) 각 노드까지의 경로들 중 패킷의 평균 왕복 시간과 패킷의 분실 수가 최소인 노드 경로를 찾기 위해 추출된 값들을 비교한다.

(5) 추출된 값을 통해 얻어진 최적의 노드 경로가 라우팅 테이블에서 최상위의 SPA 객체 참조자에 해당하는 노드 경로인지 검사한다.

(6) 만약, 최상위의 SPA 객체 참조자에 해당하는 노드까지의 경로가 최적이 아닌 경우, 패킷 왕복 시간과 패킷 분실 수의 값이 최소인 노드 경로들 순으로 라우팅 테이블을 재구성한다. 이 때, 패킷의 평균 왕복 시간 및 분실된 패킷의 수를 이용한 값 비교 시 패킷 분실 수가 적은 것을 우선으로 한다.

(7) 이동 에이전트는 라우팅 테이블의 최상위 노드를 선택하여 에이전트를 이주시킨다.

(그림 6)은 이러한 최적의 이주 경로 탐색 및 경로 조정 방식을 수행하는 과정에 대한 예이다.

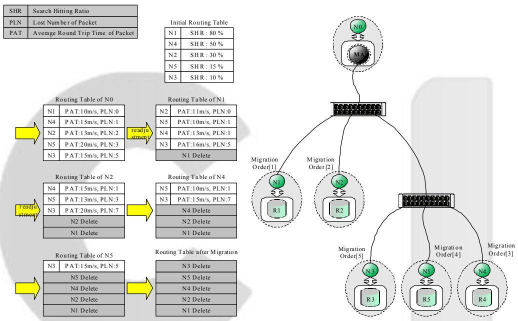


그림 6. 최적 이주 경로 탐색 및 조정 방식의 수행 과정 예
Fig 6. Optimal migration path searching and adjustment

최적 이주 경로 탐색 및 조정 방식의 수행 과정 중 최적 경로 탐색은 위의 (2)~(5)과정으로, Ping을 통해 얻어진 패킷의 평균 왕복 시간과 패킷 분실 수의 값을 비교하여 라우팅 테이블에서 최상위의SPA 객체 참조자에 해당하는 노드까지의 경로가 최적인지를 판단하는 과정이다. 한편, 최적 경로 조정 과정은 (6)과정으로, 최상위의 SPA 객체 참조자에 해당하는 노드 경로가 최적이 아닌 경우에만 패킷의 평균 왕복 시간과 패킷 분실 수의 값이 최소인 노드 경로들 순으로 라우팅 테이블을 재구성한다. 이 때, 패킷의 평균 왕복 시간 및 분실된 패킷의 수를 이용한 값 비교 시 패킷 분실 수가 적은 것을 우선으로 라우팅 테이블을 재구성하게 되는데, 이는 통신망의 결손 발생으로 인한 이동 에이전트의 이주 보장을 위해서이다.

### 4.2 노드 장애에 따른 노드 재지정 기법

이동 에이전트 시스템에서 라우팅 테이블에 따라 이동 에이전트가 이주를 수행할 때, 임의의 목적 노드까지의 경로가 외부적인 요인으로 인해 결손되어 에이전트가 이주할 수 없거나 혹은 목적 노드 자체의 장애로 인해 이미 이주하여 실행 중인 이동 에이전트가 무한 대기 또는 파괴될 수 있다. 먼저, 통신망의 결손으로 인해 에이전트가 이주하지 못하는 경우에는 3.3절에서 설명한 바와 같이 Ping의 실행 정보 중 분실된 패킷의 수인 Lost 값을 추출하여 이 값이 최소인 경로로만 에이전트의 이주를 수행함으로써 이주 신뢰성을 보장한다. 한편, 노드의 장애로 인해 이미 이주하여 실행중인 에이전트가 무한 대기 상태에 빠지거나 쓰레기로 처리되는 경우에는 노드 재지정 방법을 사용하여 해결할 수 있다. 노드 재지정 방법은 장애가 발생한 노드의 이전 이주 노드에서 새로운 이주 노드를 재지정하여 에이전트를 이주시키는 방식이다. 노드 재지정 방식의 수행 과정은 다음과 같다.

(1) 이동 에이전트는 이미 이주하여 작업을 수행한 A 노드로부터 라우팅 테이블에서 최적 이주 노드인 B 노드로 이주 한다. 이 때, B 노드로 이주하는 이동 에이전트는 자신을 복제하여 A 노드에 남겨둔다.

(2) A 노드에 복제되어 남겨진 이동 에이전트는 B 노드로부터 복제 에이전트 삭제 신호가 수신될 때까지 일정한 시간 동안 대기한다.

(3) 만약, B 노드에 장애가 발생하여 A 노드로부터 이주한 에이전트가 고아 상태에 빠지거나 파괴되었을 경우, A 노드의 복제 에이전트는 정해진 대기 시간이 지나면 B 노드에서 장애가 발생했다고 판단하고 라우팅 테이블의 우선 순위에 따라B 노드가 아닌 다음의 최적 노드인 C 노드로 에이전트를 재이주시킨다.

(4) B 노드에서 장애가 발생하지 않고 이주된 이동 에이전트가 주어진 작업을 완전하게 처리하면, B 노드의 이동 에이전트는 다음 노드로 이주하기 전 A 노드에 남겨놓은 복제 에이전트를 삭제하도록 신호를 보낸다.

(5) A 노드는 B 노드의 이동 에이전트로부터 복제 에이전트의 삭제 신호를 수신하고 남겨진 에이전트 객체를 삭제한다.

노드 재지정 방법은 이러한 과정을 통해 이주 노드들의 장애에 능동적으로 대처함으로써 보다 신뢰할 수 있도록 이동 에이전트의 이주를 보장한다. (그림 7)은 노드 재지정 방식의 수행 과정에 대한 예이다.

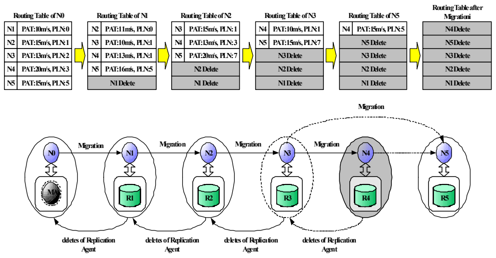


그림 7. 노드 재지정 방식의 수행 과정
Fig 7. Node assignment

## Ⅳ. 결론 및 향후 과제

본 논문에서는 자율적인 작업 처리 능력을 가진 이동 에이전트가 분산된 노드들을 이주하며 작업을 처리할 때 최소의 네트워크 지연 시간을 갖도록 라우팅 스케줄을 계획하여 노드 간 이주를 수행함으로써 보다 효율적으로 작업을 처리할 수 있도록 하기 위한 노드 간 이주 기법들을 제시하였다. 또한, 제안된 이주 기법들에서 필요로 하는 네트워크 트래픽 측정 및 네트워크 장애 검출을 위해, Ping 프로그램을 이용하여 패킷의 평균 왕복 시간과 패킷 분실 수에 대한 정보들을 획득함으로써 트래픽 크기 측정 및 통신망 결손에 대응하는 방법을 제시하였다. 최적 이주 경로 탐색 및 조정 기법은 네트워크 상에서 발생한 과다한 트래픽 경로를 회피하는 방식으로, 네트워크의 트래픽을 감지하여 최적 경로로의 이주를 수행함으로써 노드 이주 시간 비용을 최소화하기 위한 이주 방법이다. 또한, 노드 장애에 따른 노드 재지정 방법은 장애가 발생한 노드의 이전 이주 노드에서 새로운 이주 노드를 재지정하여 에이전트를 이주시키는 방식으로, 보다 신뢰할 수 있는 에이전트의 이주를 보장하기 위한 방법이다. 이러한 노드 간 이주 기법들을 통해 보다 빠르고 안정적인 이동 에이전트의 이주를 수행함으로써 효과적으로 사용자 요구사항을 처리하여 이를 서비스할 수 있다.

향후 연구 과제로는 제안된 최적 이주 경로 탐색 및 조정 기법과 노드 재지정 기법들에 대한 알고리즘의 개발이 요구되며, 이러한 노드 간 이주 기법을 이용하여 보다 효율적으로 작업을 처리하는 이동 에이전트 모델의 개발이 필요하다.

## 참고문헌

[1] K. A. Baharat and L. Cardelli, "Migratory Applications", Proceedings of the 8th annual

ACM symposium on User interface and software technology, November 1995.
[2] J. Vitek and Christian Tschudin, "Mobile Object Systems:Towards the Programmable Internet", Springer-Verlag, April, 1997.
[3] OMG, "Agent Technology Green Paper", Agent Platform Special Interest Group, http://www.objs.com/agent/ index.html, 2000.
[4] J. Baumann, "A Protocol for Orphan Detection and Termination in Mobile Agent Systems", TR-1997-09, Stuttgart University, 1997.
[5] Robert S. G, "Agent Tcl: A flexible and secure mobile-agent system", TR98-327, Dartmouth Col. June, 1997.
[6] D. B. Lange and M. Oshima, "Programming and deploying Java Mobile Agents with Aglets", Addison Wesley Press, 1998.
[7] B. Venners, "The Architecture of Aglets", JavaWorld, 1997.
[8] K. Koukoumpetsos and N. Antonopoulos, "Mobility Patterns: An Alternative Approach to Mobility Management", Pro. the 6th World Multi-Conference on Systemics, pp.14-18, 2002.
[9] Christian Erfurth, Peter Braun and Wilhelm Rossak, "Some Thoughts on Migration Intelligence for Mobile Agents", Technical Report No. 09/01, Computer Science Department, 2001.
[10] ObjectSpace, "ObjectSpace Voyager, GeneralMagic Odyssey, IBM Aglets : A Comparison", VoyagerTM, 1997.
[11] 전병국, "Design and Implementation an efficient migration policy for mobile agent", 한국정보처리학회 논문지, Vol. 6-7, pp.1770-1776, 1999.
[12] A. Yariv, D. B. Lange, Agent Design Patterns : Elements of Agent Application Design, Second International Conference on Autonomous Agents, Agents 98, 1998.
[13] 박홍진, " Design of Agent Migration Information System for Providing Optimal Achievement Order", 한국정보처리학회 논문지, Vol. 9-4, pp. 345-357, 2002.

## 저 자 소 개

**김 광 종**
1998년 군산대학교 컴퓨터정보과 학과(학사)
1999년 군산대학교 대학원 컴퓨터정보과학과(이학석사)
2004년 군산대학교 대학원 컴퓨터정보과학과(이학박사)
<관심분야> 이동 에이전트, 분산객체시스템, 능동 데이터베이스

**고 현**
2001년 군산대학교 컴퓨터정보과학과(학사)
2003년 군산대학교 대학원 컴퓨터정보과학과(이학석사)
2005년 군산대학교 대학원 컴퓨터정보과학과(박사과정)
<관심분야> 에이전트, 분산객체시스템, 멀티미디어 데이터베이스

**김 영 자**
1993년 군산대학교 컴퓨터정보과학과(학사)
2000년 순천대학교 대학원 컴퓨터정보과학과(이학석사)
2005년 군산대학교 대학원 컴퓨터정보과학과(박사수료)
2003년~현재 인천기능대학 교수
<관심분야> 에이전트, 분산객체시스템, CDN

**이 연 식**
1982년 전남대학교 전자계산학과(학사)
1984년 전남대학교 대학원 전자계산학과(이학석사)
1994년 전북대학교대학원 전산응용공학과(공학박사)
1995년~1997년 군산대학교 교무부처장
1997년~1998년 University of Missouri 교환교수
1999년~2001년 군산대학교 전자계산소 소장
1998년~현재 군산대학교 컴퓨터정보과학과 교수
<관심분야> 번역기 이론, 객체지향시스템 능동시스템, 지능형에이전트